Final Report for Contract N00014-86-K-0465

"Interactive Modeling Physical Objects"

Christoph M. Hoffmann
Computer Science Department
Purdue University
West Lafayette, Indiana 47907
(317)-494-6185

Contract Period of August 1986 through January 1990

1 Summary of Activities

The two foci of work have been the Newton project, simulating mechanical systems of rigid bodies, and geometric modeling research, studying how to represent physical shapes, how to query such representations, and how to manipulate them. The Newton work matured from the basic system design to a sophisticated simulation system with a strong user interface and substantial physical and geometric capabilities. For technical details see Section 2.1. The geometric modeling research explored the utility of methods from algebraic geometry, numerical analysis, and differential geometry concepts. To a large part, this research has been synthesized into the monograph *Geometric and Solid Modeling* [1]. For technical details see Section 2.2.

The books, reports and papers completed during the reporting period are listed below in Section 2.4. I joined the editorial board of the "Journal of Symbolic Computation," the "Journal for Applicable Algebra," and the journal "Computer-Aided Geometric Design." Moreover, I organized a number of courses and workshops, and made contributed and invited presentations at many other conferences and workshops. See also Section 2.3.

2 Technical Details of the Work

2.1 Project Newton

The objective of this project is the creation of a system for simulating the dynamical behavior of mechanisms and mechanically evolving environments. Such

a system can be used in mechanical engineering, to analyze the behavior of mechanisms, it can be used in robotics, to verify off-line programs controlling robotic manipulators, and it can be used in manufacturing, to prototype products and analyze them.

The system was conceived in the Spring of 1986 in collaboration with John Hopcroft while I was visiting Cornell University, [2]. Since this collaborative design, both the evolution and the implementation of the system have been coordinated. Over the contract period, an informal division of labor developed, with the Cornell group doing substantial work on control algorithms, while the Purdue group focused on expanding the physics and the geometric coverage.

The main technical objectives of the system design include allowing unpredicted and unpredictable collisions and contacts between bodies, and to direct the elaboration of events completely automatically through the geometry of the objects in the simulated system. See also [3]. Moreover, the design called for a simple and natural user-interface to make the system accessible to nonspecialists.

The first system design and implementation restricted the geometry of objects to articulated cuboidal shapes so as to concentrate fully on the physics. After implementing Newton-Euler dynamics, collisions were added using the accepted model of impulsive force action governed by a coefficient of restitution that is material-specific and can be altered as desired. Contact elaboration proceeds by monitoring all contact forces and structurally altering the system of differential equations that governs motion. All such changes are completely automatic and do not require any user input.

Shortly after the system came on-line, early Fall of 1986, simple mechanisms were simulated and control algorithms to manipulate linkages were developed in an effort to assess the flexibility and soundness of the design. During 1987, it was felt that a more flexible dynamics engine would be desirable, so that the system could be run both in forward-dynamics and inverse-dynamics modes. This system reorganization was undertaken 1988 with a complete rewrite that enabled the system to determine automatically whether to determine forces from accelerations, or to determine accelerations from forces. This objective was achieved by adding symbolic computation capabilities to the system that effectively compile motion equation schemata into the final system of ODEs.

At the time of this rewrite, the geometric capabilities were still quite limited. Late 1988, the Purdue team experimented interfacing Newton's rigid-body dynamics with finite-element codes. An experimental interface to DLEARN was created, and Newton's geometry engine was augmented to understand flexible objects consisting of hexahedral finite elements.

During the Fall of 1989, Dr. George Vaněček joined our department as research associate and has been collaborating with me. Vaněček had designed and implemented the polyhedral solid modeling system *ProtoSolid* for his PhD at

the University of Maryland, under the direction of Dana Nau. After joining the Purdue team, Vaněček integrated ProtoSolid into Newton. The incorporation of ProtoSolid into Newton has four technical aspects.

1. Implement basic communication protocols and interfaces between Newton and ProtoSolid.
2. Implement mass property calculations in the geometric modeler so that Newton can work with all solids defined in ProtoSolid.
3. Devise data structures to support efficiently geometric collision detection and analysis.
4. Devise methods to increase tne robustness of processing collisions and temporary contacts over time, and preventing the accumulation of positional errors between contacting bodies.

During the reporting period, the first two points were completed. The other two points are the subject of research conducted later and sponsored under a renewed ONR contract.

In addition to integrating ProtoSolid, Vaněček developed a sophisticated graphical user interface that runs on a front-end graphics workstation. The user interacts first with the solids modeler to define graphically the component shapes. Then, the shapes are imported into Newton, positioued relative to each other, and linked with hinges. Thereafter, the simulation can be initiated. Throughout the interaction, the interface tool compiles a textual protocol of the session, so that definitions can be archived and reused.

2.2 Geometric Modeling Research

Early geometric modeling work has explored techniques for blending surfaces. Given two surfaces, a blending surface is a third surface that smoothly joins the given ones. Blending surfaces are on virtually all man-made objects, and are variously called rounds, fillets, and so on. While such surfaces are easily understood intuitively, specifying them with precision is not a simple matter in general. In joint work with John Hopcroft I have devised a substitution technique for deriving blendiug surfaces for given implicit surfaces, [4, 5].

In 1986, I was involved with John Hopcroft and Michael Karasick in the development of a polyhedral solid modeling system. The implementation of the system raised fundamental issues of robustness. Briefly, geometric algorithms are designed with exact arithmetic in mind, but are implemented using floating-point arithmetic. Normal errors of round-off and digit cancellation result in geometric tests that can yield contradictory answers and then lead to program failure. In the modeler, we developed a reasoning strategy based on

prior geometric tests and the current computation to overcome incorrect answers. We call this approach to increasing robustness the *reasoning paradigm*; [6, 7]. Other strategies are possible; for surveys of current techniques see [8] and [1, Chap 4].

When curved surfaces are allowed, robustness is not the only problem that must be addressed by solid modeling systems. Customary data structures to represent the solid boundary (Breps) do not necessarily contain all information needed to reconstruct the solid unambiguously, [9]. The problem arises in the edge definition. Essentially, edges are segments of surface intersection curves, and when the intersection curves are singular at a vertex, ambiguities are possible. In [10], we have considered how to augment traditional surface intersection evaluation methods with techniques from algebraic geometry. Specifically, the intersection is mapped to a plane curve. When approaching a singularity, the curve is subjected to a quadratic transformation whose effect is to structurally simplify the singularity. A theorem from advanced algebraic geometry states that repeated applications of quadratic transformations eliminates the singularity completely. We implemented this strategy, marching on the given curve where it is not singular, and traversing the desingularized curve across the singularities of the original curve; [10, 11].

While quadratic transformations are simple in nature and straightforward in application to practical problems in geometric modeling, other techniques originating from algebraic geometry may involve large symbolic computations that can be impractical. In particular, converting from parametric to implicit surface representation, while always possible in principle, is often too expensive. Together with my student J.-H. Chuang, I have explored cheaper local methods that approximate parametric curves and surfaces with implicit ones, [12]. As the accuracy of the approximation increases, the implicit approximant converges to the exact implicit form. In principle, this numerical method is an attractive alternative to the expensive symbolic computation deriving the implicit form.

In the work on surface intersection evaluation [10], a basic tracing algorithm was developed that is also capable of evaluating the intersection of parametric surfaces. In assessing this algorithm, I realized that the method really constructs local approximants for 1-manifolds in higher-dimensional space. Moreover, when representing constrained surfaces explicitly as 2-manifolds in higher-dimensional space, it becomes straightforward to implement surface operations such as offsets and spherical blends. Specifically, when defining constrained surfaces mathematically, the definition simplifies if one allows it to be made in a higher dimensional space. The extra dimensions are auxiliary variables that express the various constraints explicitly. I have called this approach a *dimensionality paradigm* since it trades off more equations against lower algebraic degrees. Thus, an offset surface can be defined by four equations whose degree is no higher than the degree of the original surface, or by a single equation,

obtained after laborious symbolic computation, of degree often eight times or more the degree of the original surface. Furthermore, such closed-form representations are often impossible to obtain in practice because the derivation involves computations of exponential complexity.

In [14], the formulation of complex constrained surfaces is presented, including offsets, equi-distance surfaces, and fixed-radius and variable-radius blends. This paper also extends the surface intersection algorithm of [10] to 2-manifold intersections in n-space. In [15] further problems related to the dimensionality paradigm are discussed, and a local approximation schema is presented. Moreover, the paper discusses how to establish a correspondence between engineering intent and mathematical problem formulation. Both [14] and [15] begin to develop an algorithmic infrastructure for manipulating surfaces defined by the dimensionality paradigm as 2-manifolds in n-space. The significance of this work is that it develops an enabling technology that puts into reach applications such as the computation of the medial-axis transform of three-dimensional domains which, in turn, should simplify problems such as finite-element mesh generation and geometric tolerancing.

2.3 Talks, Workshops and Conferences

In 1987, I organized a minisymposium on blending surfaces for the SIAM Conference on Geometric Modeling and Robotics, in Albany. The speakers included John Owen (Shape Data Ltd), Alyn Rockwood (Evans and Sutherland) and Lasse Holmstrom (University of Helsinki). I also organized a summer program on Computational Issues in Robotics for the Institute for Mathematics and Applications at the University of Minnesota. The principal speakers in this program included Bruno Buchberger (RISC Linz), Kokichi Sugihara (University of Tokyo), and Deepak Kapur (General Electric). During 1987, I gave invited talks at the IMA workshop on Supercomputing in Minnesota, and at the NSF Research Conference on Geometric Modeling in Detroit.

In 1988, I organized a SIGGRAPH Course on Algebraic Geometry jointly with S. Abhyankar and C. Bajaj. I also organized a workshop on Algorithmic Aspects of Geometry and Algebra, jointly with C. Yap and E. Kaltofen. The workshop was sponsored by the Army's Mathematical Sciences Institute at Cornell. Among the keynote speakers were Shreeram Abhyankar (Purdue), Bruno Buchberger (RISC Linz), George Collins (Ohio State), John Hopcroft (Cornell), and Wu Wen-Tsun (Peking). That year, I served on the NSF panel evaluating small-scale infrastructure proposals, as well as on the NASA panel reviewing the CESDIS grant applications. I was site visitor evaluating Rochester's CER grant application. In 1988 I accepted invitations to speak at the NATO workshop on CAD-based Programming for Sonsory Robots in Il Ciocco, Italy; at the NSF-IFIP workshop on Geometric Modeling at Rensselaerville; and at the MSI workshop on Gröbner Bases in Cornell.

In 1989, I organized two minisymposia for the SIAM Conference on Geometric Design in Tempe, one on Computing about Physical Objects, the other on Accuracy and Robustness in Geometric Computations. The speakers included Jim Cremer (Cornell), Demetri Terzopoulos (Toronto), Joe Thompson (Mississippi State), Peter Kahn (Cornell), Kokichi Sugihara (Tokyo), and Leo Guibas (Stanford). I accepted invitations to speak at the Conference on Applicable Algebra, and the Conference on Surfaces in Geometric Computations, both held in Oberwolfach, Germany. I also accepted an invitation to lecture at the NATO ASI Seminar in the Canary Islands. That year, I served on the program committee of the ACM conference on Computational Geometry, and served on the review panel for the ONR URI program at the University of North Carolina.

2.4 Reports and Publications

1. *Geometric and Solid Modeling, An Introduction*, Morgan Kaufman Publishers, San Mateo, Cal., 1989.

2. "Simulation of Physical Systems from Geometric Models," *IEEE Journal on Robotics and Automation RA-3*, 1987, 194–206; (with J. Hopcroft).

3. "Model Generation and Modification for Dynamic Systems from Geometric Data," in *CAD Based Programming for Sensory Robots*, B. Ravani, ed., Springer NATO ASI Series F-50, 481–492, 1988; (with J. Hopcroft).

4. "The Potential Method for Blending Surfaces and Corners," in *Geometric Modeling*, G. Farin, ed., SIAM, 1987, 347–365; (with J. Hopcroft).

5. "Projective Blending Surfaces", *Artificial Intelligence 37*, 1988, 357–376; (with J. Hopcroft).

6. "Towards Implementing Robust Geometric Computations," 5^{th} *ACM Symp. Comp. Geometry*, Urbana, Ill., 1988; (with J. Hopcroft and M. Karasick).

7. "Robust Boolean Operations on Polyhedral Solids," *IEEE Trans. Graphics 9*, 1989, 50–59; (with J. Hopcroft and M. Karasick).

8. "The Problems of Accuracy and Robustness in Geometric Computation," *IEEE Computer 22*, 1989, 31–42.

9. "Geometric Ambiguities in Boundary Representations," *Computer Aided Design 19*, 1987, 141–147; (with J. Hopcroft).

10. "Tracing Surface Intersections," *Computer-Aided Geometric Design 5*, 1988, 285–307; (with C. Bajaj, J. Hopcroft, and R. Lynch).

11. "Algebraic Curves," in *Mathematical Aspects of Scientific Software*, J. Rice, ed., IMA Volumes in Math. Applic., Springer Verlag, 1988, 101–122.

12. "Local Implicit Approximations of Curves and Surfaces," *ACM Trans. on Graphics 8*, 1989, 298–325; (with J.-H. Chuang).

13. "On the Geometry of Dupin Cyclides," *The Visual Computer 5*, 1989, 277–290; (with V. Chandru and D. Dutta).

14. "A Dimensionality Paradigm for Surface Interrogation," *Computer Aided Geometric Design 7*, 1990, 517–532.

15. "Algebraic and Numerical Techniques for CAGD," in *Computations of Curves and Surfaces*, W. Dahmen, M. Gasca, C. Micchelli, eds., NATO ASI Series C, Vol. 307, Kluwer Academic, London 1990, 499–528.